
Experience With Multiple FOMs

Russ Richardson

AMG 18, April 9, 1997

FOM Development Process

- Started with existing well-defined scenario
 - Translated JPSPD Interface Requirement Specification (IRS) to OMT format
 - Refined JPSPD Interest Mgmt. scheme (multicast groups) to HLA IM scheme
- Tools
 - Manual entry into Excel Workbook. Tedious and hard to maintain due to multiple views of the same data. (Majority of time spent here)

Resulting Product

- DIS-like Data Representation

Entity	Platform	Land	Tank	M1
				T72
				T54
			ArmoredFightingVehicle	BMP - 1
				BTR80
		Air	SelfPropelledArtillery	M270_ATACMS
				M109
			SmallWheeledUtilityVehicle	M577A1
			AttackHelicopter	AH64
				RAH66
	Munition		ElectronicWarfare	JSTARS
			UAV	HUNTER_2GEN
			Guided	BAT_P3I
		AntiArmor		
		BattlefieldSupport9	ATACMS_MISSILE	

- Class attributes are minimal fields of EntityState PDU for each entity type
- Interactions are used for sporadic PDUs, Tactical Messages, hand-off to engineering models, and Aggregation/Disaggregation

Resulting Product (cont.)

- Component Table specifies mapping between Aggregate and Entity representation (specifies ModSAF CLCGF template definitions)
- Data structure table defines complex attributes

RED_TANK_CO [9]	T54 [10]
	BTR80 [3]
RED_TB_PLUS [18]	BMP-1 [10]
	T72 [30]
BLUE_MECH_DIV_CP [1]	M1 [5]
	M577A1 [12]

DataStructure	Field	Datatype
RE_Reference	Title	string
	Originator	string
	Day	short
	Hour	short
	Minute	short
	SerialNumber	string
	SpecialNotation	string
	NASIScode	string
	Ampn	string
	Narr	string

Interoperability

FOM Development Process

- Started with existing CCTT and TestBed FOM
 - Agreed on a merged FOM to minimize development
 - Naming Convention and class structures defined
 - Used DIS 2.04 enumerations and PDU contents to form basis for attributes and classes
- Tools
 - Used Aegis FOM development tool

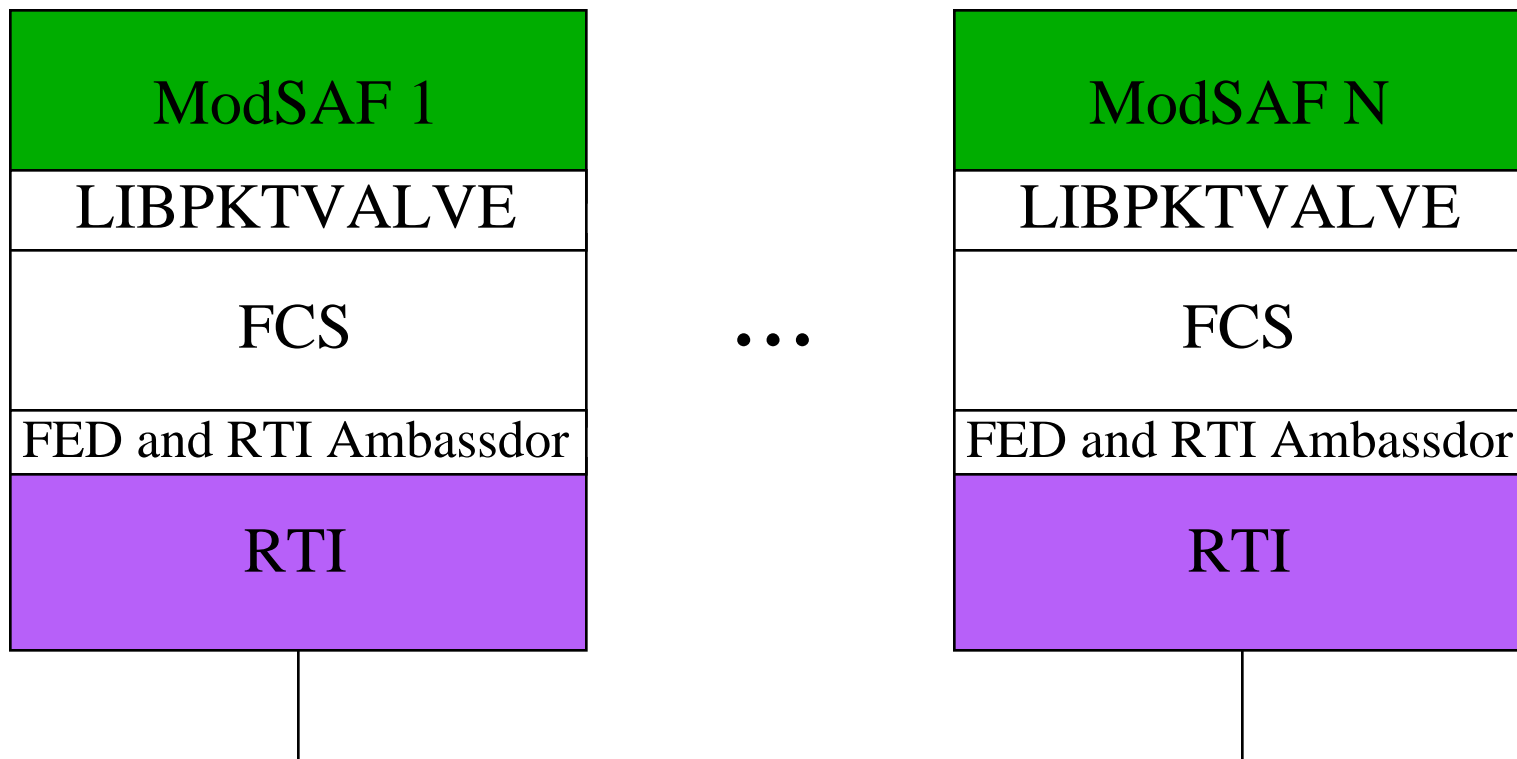
Interoperability FOM

Entity	M1
	T72
	M2
	BMP
	FA 18
	Infantry
	TOW
	AT 5
	SA 16
	LGB

- **Flat Structure** - all attributes are specified at the Entity level
- **Sub-Classes** are defined for class based filtering at Entity type level

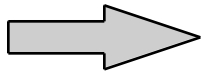
HLA ModSAF Architecture

- Each ModSAF Federate is an Executable



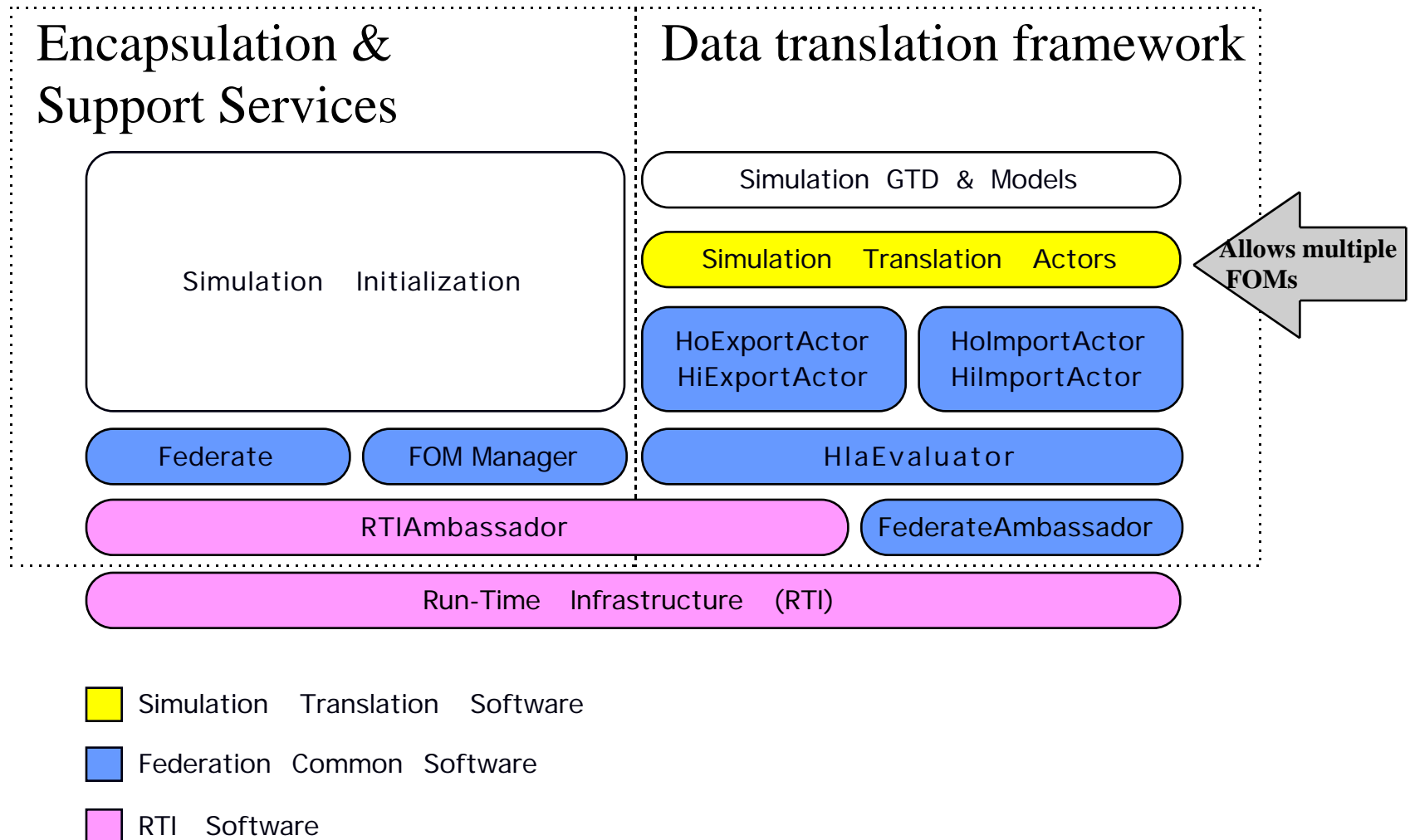
Federation Common Software (FCS)

- Purpose: Develop software to facilitate the integration of CLCGF & HLA Testbed simulations with the RTI.
 - Encapsulation and automation of services all simulations must exercise (create/destroy/join/resign federation, publication/subscription, etc.)
 - FOM Management and RTTI services
 - Support for OO FOM data representation (deep class hierarchy etc.)
 - Efficient mapping between RTI Run-time typing and simulation compile-time typing

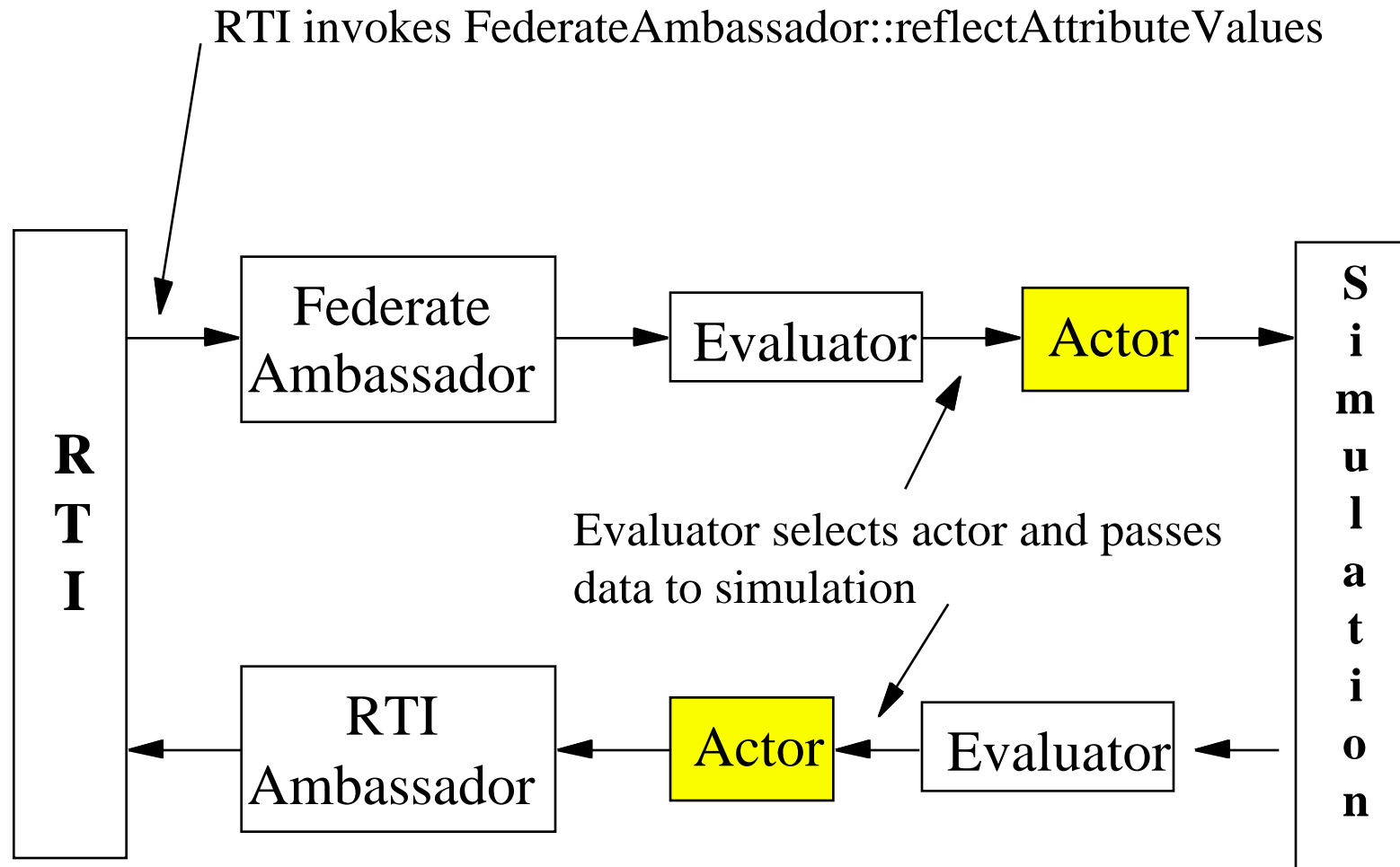


- **Framework for translation between simulation and FOM data representation**
- Common instrumentation for performance analysis

Federation Common Software



Data Translation Flow



Actors are FOM specific

Modifications to FCS to Support Second FOM

- What was needed:
 - Means of registering/subscribing for data at subclass level
 - CCTT supported subclasses to the object class Entity, but the FCS did not. (The implemented FCS actors were defined for the STOW FOM.) Minor code development necessary to support subscription and registration services for objects.
 - Develop Actors to handle remapping of munition enumeration
 - Since Both FOMs were DIS based (the data that the simulations needed to communicate are essentially the same), the biggest issue was agreeing on a common representation.
 - This took a couple of meetings- we had several misunderstandings that didn't get worked out until testing (e.g. different interpretations for the type of OMT enumeration).
- Issues for Additional FOMs (e.g. RPR FOM)
 - Mechanism to support complex data types (common marshalling)
 - Actors for non-DIS type interactions and base classes